Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-17

THE BERTH ALLOCATION PROBLEM: APPLICATION TO THE GIOIA TAURO MARITIME TERMINAL

by

Jean-François Cordeau

Gilbert Laporte

Pasquale Legato

Luigi Moccia

August 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

103-07112/

PUBLICATION
CRT-2003-17

THE BERTH ALLOCATION PROBLEM: APPLICATION TO THE GIOIA TAURO MARITIME TERMINAL

by

Jean-François Cordeau[1]

Gilbert Laporte[1]

Pasquale Legato[2]

Luigi Moccia[1,2]

[1] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 and Canada Research Chair in Distribution Management, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[2] Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS), Italy

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2003

Abstract

In the *Berth Allocation Problem* (BAP) the aim is to optimally schedule and assign ships to berthing areas along a quay. The objective is the minimization of the total service time for all ships, defined as the time elapsed between the arrival in the harbour and the completion of handling. Two versions of the BAP are considered: the discrete case and the continuous case. The discrete case works with a finite set of berthing points. In the continuous case ships can berth anywhere along the quay. Two formulations and a tabu search heuristic are developed for the discrete case. Only small instances can be solved optimally. For these sizes the heuristic always yields an optimal solution. For larger sizes it is always better than a truncated branch-and-bound applied to an exact formulation. A heuristic is also developed for the continuous case. Computational comparisons are performed with the first heuristic and with a simple constructive procedure.

Keywords: berth allocation, maritime container terminal, vehicle routing, scheduling, tabu search, time windows.

Résumé

Cet article porte sur un problème d'accostage des navires dans lequel on doit déterminer le moment et l'endroit ou des navires accosteront le long d'un quai. Chaque navire a une heure d'arrivée connue et son temps de traitement dépend de sa position le long du quai. L'objectif consiste à minimiser la somme des temps d'attente et de traitement des navires. Deux versions du problème sont considérées ; le cas discret et le cas continu. Dans le cas discret, on considère un ensemble fini de points d'accostage possibles alors que le cas continu suppose que les navires peuvent accoster en tout point le long du quai. Deux formulations ainsi qu'une méthode de recherche avec tabous sont développées pour le cas discret. Une heuristique est également proposée pour le cas continu. Des résultats numériques sont finalement rapportés.

Mot-clés : opérations d'accostage, port de conteneurs, tournées de véhicules, horaires, recherche avec tabous, fenêtres de temps.

1 Introduction

There are more than 2000 ports around the world, ranging from single berth locations handling a few hundred tons a year to multipurpose facilities handling up to 300 million tons a year. According to a report of the World Bank (2003), the expected annual increase in world port traffic is between 4 and 5%. Today more than 80% of the trade with origin or destinations in developing countries is waterborne. In the last 20 years the use of containers for general cargo has increased steadily. Containers are large metal boxes made in standard dimensions and measured in multiples of twenty feet called "twenty foot equivalent units" (TEUs). Between 1990 and 1998, container traffic has doubled worldwide, reaching 175 million TEUs. The forecast is for 270 million TEUs by 2005.

In maritime container transportation the hub and spoke arrangement is widely adopted. Deep sea vessels, also called mother vessels, operate between a limited number of transshipment terminals (hubs). Smaller vessels (feeders) link the hubs with the other ports (spokes). This network topology results in the consolidation of capacity along the routes linking the transshipment ports and in the growth of their importance. In recent years, mother vessels have strongly increased in size attaining up to 6000 TEUs. Transshipment ports are large intermodal platforms and a limited number of them handle an important share of the world traffic. Thus, in 1998 the first ten container ports handled 40% of the total traffic. Ultra-large container vessels cut down transport cost. However, hub ports are forced to invest heavily to accommodate these ships by deepening and widening channels and constructing new berthing facilities of sufficient depth and length. These trends require a continuous improvement in managerial practices at transshipment terminals which can be viewed as material handling systems.

When ships arrive at the port, they enter in the harbor waiting to moor at the quay (Figure 1). The quay is a platform protruding into the water to facilitate the loading and unloading of cargo. The locations where mooring can take place are called berths. These are equipped with giant cranes, called pier or quay cranes, used to load and unload containers which are transfered to and from the yard by a fleet of vehicles. In a transshipment terminal the yard allows temporary storage before containers are transfered to another ship or to another mode (e.g., rail or road). Containers relocation in the yard is also performed to speed up the loading process. A high degree of coordination is required to ensure a fast transshipment process. Advanced communication and information technologies already exist and the next step will be the introduction of customized optimization techniques. Customization is a key factor since maritime terminals differ from each other in their layouts and material handling equipments. The need for an optimal management of logistic activities at modern container terminals is well recognized (Daganzo, 1990; De Castilho and Daganzo, 1993; Taleb-Ibrahimi et al., 1993). For a recent overview and classification of the various equipments and decision problems in such

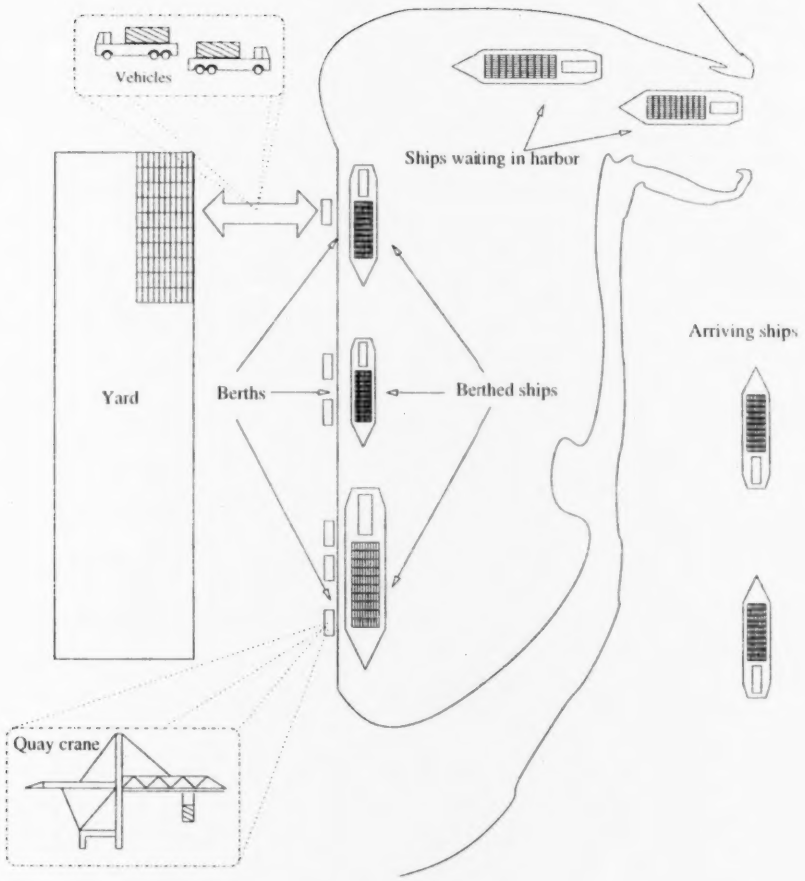

Figure 1: Maritime terminal layout

systems, see Vis and De Koster (2003). A review of operational research issues in maritime logistics, focused on ship routing and scheduling, is presented in Christiansen et al. (2003).

1.1 Overview of the Berth Allocation Problem

The berth allocation problem consists of assigning incoming ships to berthing positions. Managers face two interrelated decisions: *where* and *when* the ships should moor. The problem can be represented in a two-dimensional space (Figure 2) where ships are rectangles whose dimensions are the ship length, including a safety margin, and the handling time. These rectangles must be placed in the decision space without overlapping each other and while satisfying several constraints.

In the spatial dimension, there are constraints relative to the depth of the water (allowable draft) and to the maximum distance in relation to the most favorable location along the quay,

computed with respect to the location of the outbound containers and to the reserved space for the inbound containers. In the temporal dimension the constraints are expressed as time windows on the completion time of ship servicing. Some time windows are soft and can be relaxed with an appropriate penalty cost.

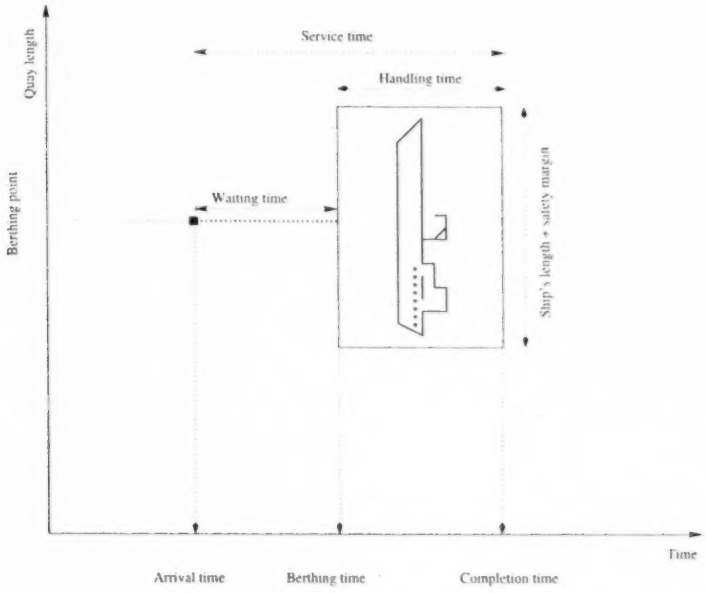

Figure 2: Berth - time space

The handling time data deserve a more detailed discussion. These data depend upon another related decision which is the number of quay cranes assigned to the incoming ships. Of course, this decision affects the handling time and has an impact on the BAP. In a complex system like a transshipment port the decision making process is often hierarchical, and the quay cranes assignment problem is solved before the berth allocation. In addition, the handling time of a ship depends on its berthing point and is a function of the distance from the berth to the pick-up and delivery area of containers stored in the yard. This dependency strongly affects the performance of the port.

The planning horizon of the BAP is one week, and the berthing plan is updated every day. Since the problem is solved with a rolling horizon some areas of the berth-time space are not always available when the problem is reoptimized. Some parts of the quay may also be unavailable because of maintenance operations. The arrival time of the ships is known in advance. Every ship has its own time window determined by its arrival and its maximal allowable completion time. Managers want to minimize both port and user costs, which are related to service time. The objective of the BAP is to minimize the total service time of all ships. In some versions of the problem it is possible to include some penalty terms in the objective function. For example, as mentioned, there can be a penalty when the service time of a ship

exceeds the contracted value.

1.2 Literature review

The BAP can be modeled as a discrete problem if the quay is viewed as a finite set of berths. In this case the berths can be described as fixed length segments, or, if the spatial dimension is ignored, as points. When ships are of different lengths, dividing the quay into a set of segments is, however, difficult to accomplish because requirements vary dynamically. Using long segments will result in poor space utilization while using short segments will likely result in an infeasible solution. Continuous models circumvent these difficulties by considering that ships can berth anywhere along the quay.

In the discrete case, the BAP can be modeled as an unrelated parallel machine scheduling problem (Pinedo, 1995) where a ship is treated as a job and a berth as a machine. The ship's arrival time is the job release time. In the continuous case, the BAP is a two-dimensional cutting stock problem with additional constraints. In both cases, the BAP is NP-hard (Garey and Johnson, 1979).

Imai et al. (2001) have proposed the *Dynamic Berth Allocation Problem* (DBAP) formulation in which the quay is represented as a finite set of berthing points. In other words, the spatial dimension of ships and berths is not considered. This formulation is called "dynamic" as opposed to a previous one called the *Static Berth Allocation Problem* (SBAP) (Imai et al., 1997) which considers the case where all ships are already in the port when the berths become available. The SBAP is solvable in polynomial time with the Hungarian method (Papadimitriou and Steiglitz, 1982) since it is reducible to an assignment problem. In their paper, the authors take advantage of this characteristic. They propose a suitable Lagrangean relaxation for the DBAP where the subproblem is an assignment problem. Their computational results show that the DBAP is easy to solve as long as the instances are "close" to the static case, in the sense that most ships are already in the port when the berths become available. The objective function is the sum of the ship service times. As the authors point out this objective function does not consider ship priorities. The DBAP formulation is further discussed in Section 2.1.

Nishimura et al. (2001) have presented a non-linear integer program and a genetic algorithm based on a different representation of the spatial dimension in which the quay is a collection of segments and up to two ships can share the same segment at the same time if their lengths are compatible with the length of the berth segment. Additional constraints relative to the water depth of the berths are also introduced.

The DBAP formulation was extended (Imai et al., 2003) to consider service priorities which are handled by introducing in the objective function a term corresponding to service time. Priorities, based for example on volumes, can also be incorporated in the model. The resulting formulation is non-linear. The authors show that with a suitable Lagrangean relaxation, the subproblem becomes a quadratic assignment problem. Since this problem is not well solved by exact methods, the authors have developed a genetic heuristic.

In Lim (1998) the quay is represented as a continuous line. A heuristic solves the problem

of deciding the berthing points given the berthing time of the ships, assuming constant handling times. This approach does not solve the general problem in which the berthing time is a decision variable and the handling time varies along the quay.

More recently Park and Kim (2003) have introduced a non-linear integer programming model that also considers quay crane assignments, but assumes linearity between the handling time and the number of cranes. Here the quay is represented as a continuous line. The objective function to minimize is the sum of penalty terms over all ships. For example, in the temporal dimension, the ship waiting times generate a linear cost for the port. In the spatial dimension, the authors assume that an optimal berthing point is known, and they apply a penalty whenever a different choice is made. The algorithm uses Lagrangean relaxation and a subgradient optimization technique.

Finally, Kim and Moon (2003) have formulated a mixed integer linear programming model for the continuous case. A commercial solver is able to find the optimum for instances involving seven ships and a three day time horizon. A simulated annealing heuristic is proposed to solve instances of realistic dimensions.

1.3 The Gioia Tauro Maritime Terminal

This study was initiated at the request of the Gioia Tauro Maritime Terminal, located in southern Italy. This port is mainly devoted to transshipment activities involving mother vessels and feeders operating in a hub and spoke manner. Nearly fifty spoke ports are linked to Gioia Tauro. In just a a few years, Gioia Tauro has become the largest transshipment port on the Mediterranean Sea. This achievement is impressive since the port only opened in 1995, and no infrastructures existed in 1993. The traffic forecast for 2003 is 3.2 million TEUs and 3250 ships. The harbor entrance is 250 meters wide and the water depth is 18 meters. The quay length is 3100 meters but only 2850 meters are relevant to our study. The channel along the quay presents a multi-water depth configuration, ranging from 13.5 to 15.5 meters. There are 23 quay cranes available, 18 of which are mounted on rail, while the others roll on tires. A fleet of 75 vehicles, called straddle carriers, transfer the containers between the quay cranes and the yard. These vehicles are capable of transporting up to two containers at a time and can insert containers directly into the right yard slot. Straddle carriers are usually used for the transport of full containers over relatively short distances (less than 500 meters). Different vehicles are used for longer transfers. The yard surface occupies 1.1 million square meters and can store nearly 50,000 TEUs (1,100 of them can be refrigerated). The storage area is divided into bays. Each bay is made up of 32 rows, each having 16 slots. Up to three containers can be stacked in each slot. Empty containers, which occupy approximately 40% of the storage area, have an eight to ten day average dwell time (much more than for a full container) and are located in the most remote positions. A railway station and a connection with the southern Italy highway network are also present. At the time of writing, Gioia Tauro employs 940 workers.

1.4 Outline of the remainder of this article

Our aim is to devise fast and effective heuristics for the discrete and continuous cases of the BAP. We first model the discrete version of the problem and we then extend it to the continuous case. In the discrete case medium size instances can be solved exactly under some assumptions, which enables an assessment of the quality of the heuristic. Since the continuous problem cannot be solved exactly, the assessment of the heuristic developed for this case can only be inferred from the discrete scenario. Two exact formulations for the discrete case will be presented and discussed in the Section 2. Tabu search heuristics for the discrete and continuous cases are introduced in the Section 3, followed by computational results in Section 4, and by the conclusions in Section 5.

2 Mathematical Models

We consider two different formulations for the discrete case. The first is the *Dynamic Berth Allocation Problem* (DBAP) proposed by Imai et al. (2001). The second is a *Multi-Depot Vehicle Routing Problem with Time Windows* (MDVRPTW) formulation.

2.1 DBAP formulation

As mentioned in the literature review, the DBAP formulation considers the quay as a finite set of berthing points. In what follows berthing points or berth segments will be called berths. This formulation takes into account the ship arrival times, which can be seen as lower bounds of the service time windows of the ships, but it does not consider upper bounds. In fact, the authors point out that their model does not consider the relative importance of the ships. In reality some ships, like mother vessels, have tight time windows. The model is based on position variables for the ships assigned to the berths. All input data for the model are deterministic:

- N: set of ships and $n = |N|$;
- M: set of berths and $m = |M|$;
- t_i^k: handling time of ship i at berth k;
- a_i: arrival time of ship i;
- s^k: start of availability time of berth k;
- e^k: end of availability time of berth k.

The model also uses the following variables and sets:

- P: set of service positions, $|P| = |N| = n$;
- $x_{ip}^k \in \{0,1\}$ $k \in M$, $i \in N$, $p \in P$; $x_{ip}^k = 1$, if and only if ship i is assigned at position p to berth k;

- $w_{ip}^k \in \Re$ $k \in M$, $i \in N$, $p \in P$: idle time of berth k between the departure of the $(p-1)^{th}$ ship and the arrival of the ship i assigned to the position p;
- $N(k) = \{i \in N \mid a_i \geq s^k\} \subseteq N$: set of ships that arrive in the port after berth k becomes available;
- $P(p) = \{q \in P \mid q < p\} \subseteq P$: set of positions before p.

The DBAP can then be formulated as follows:

$$\text{minimize} \sum_{k \in M} \sum_{i \in N} \sum_{p \in P} \{(n-p+1)t_i^k + s^k - a_i\} x_{ip}^k + \sum_{k \in M} \sum_{i \in N(k)} \sum_{p \in P} (n-p+1) w_{ip}^k \quad (1)$$

subject to

$$\sum_{k \in M} \sum_{p \in P} x_{ip}^k = 1 \qquad \forall i \in N \quad (2)$$

$$\sum_{i \in N} x_{ip}^k \leq 1 \qquad \forall k \in M,\ p \in P \quad (3)$$

$$\sum_{j \in N} \sum_{l \in P(p)} (t_j^k x_{jl}^k + w_{jl}^k) + w_{ip}^k - (a_i - s^k) x_{ip}^k \geq 0 \qquad \forall k \in M,\ i \in N(k),\ \forall p \in P \quad (4)$$

$$\sum_{i \in N} \sum_{p \in P} (t_i^k x_{ip}^k + w_{ip}^k) \leq e^k - s^k \qquad \forall k \in M \quad (5)$$

$$x_{ip}^k \in \{0, 1\},\ w_{ip}^k \geq 0 \qquad \forall i \in N,\ k \in M,\ p \in P. \quad (6)$$

In this model, constraints (2) state that every ship must be assigned to only one berth. Constraints (3) allow at most one ship to be assigned to the same position on the same berth. When a ship i is assigned to berth k in position p, and its arrival time a_i is larger than the completion time of the ship served in position $(p-1)$, then some idle time w_{ip}^k is incurred at the berth. Constraints (4) impose conditions on w_{ip}^k. If ship i is assigned to position p at berth k and has an arrival time a_i less than the completion time of the ship in position $(p-1)$, then w_{ip}^k is zero. Otherwise the objective function (to be minimized) forces w_{ip}^k to be equal to a_i minus the completion time of the ship in position $(p-1)$. We have added constraints (5) to the original formulation in order to take in account the available time of the berths.

The objective function minimizes the sum for each ship of the service time. As mentioned, the service time is the difference between the completion time and the arrival time of the ship in port. Imai et al. (2001) have proved that in an optimal solution q ships assigned to the same berth k are scheduled consecutively from position $(n-q+1)$ to n, meaning that $x_{ip}^k = 0 \, \forall i \in N,\ p \leq n-q$. The completion time of a ship i, assigned to berth k in position p, can be obtained by adding:

- s^k: start of availability time of berth k;
- the handling times of the ships served before the p^{th} position on the berth: $\sum_{j \in N} \sum_{l \in P(p)} t_j^k x_{jl}^k$;
- the berth idle times caused by the ships served before the p^{th} position on the berth: $\sum_{j \in N} \sum_{l \in P(p)} w_{jl}^k$;
- $t_i^k + w_{ip}^k$, the sum of the handling time of the ship i and the berth's idle time between the departure of ship $(p-1)$ and the arrival of ship i.

Subtracting the arrival time from the completion time we obtain the service time. This yields the following non-linear objective function to be minimized:

$$\sum_{k \in M} \sum_{i \in N} \sum_{p \in P} \left\{ \sum_{j \in N} \sum_{l \in P(p)} (t_j^k x_{jl}^k + w_{jl}^k) + w_{ip}^k + t_i^k + s^k - a_i \right\} x_{ip}^k. \tag{7}$$

The following observation can be used to linearize this function. Every handling time t_i^k, as well as each idle time w_{ip}^k, are part of the completion time of the ship i assigned to berth k, but are also part of the completion time of the $(n-p)$ ships scheduled after i. Therefore, (7) can be written as:

$$\sum_{k \in M} \sum_{i \in N} \sum_{p \in P} \left\{ (n-p+1)(w_{ip}^k + t_i^k) + s^k - a_i \right\} x_{ip}^k. \tag{8}$$

It can be seen that the objective function (1) used by Imai et al. is a refinement of (8).

2.2 MDVRPTW formulation

Legato et al. (2001) have observed that the BAP can be modeled as an MDVRPTW (see, e.g., Cordeau et al., 2001). In this model the ships are seen as customers, and the berths as depots at which one vehicle is located. There are then m vehicles, one for each depot. Each vehicle starts and ends its tour at its depot. The ships are modeled as vertices in a multi-graph. Every depot is divided into an *origin* vertex and a *destination* vertex. Time windows can be imposed on every vertex. At the origin and destination vertices, the time windows correspond to the availability period of the corresponding berth. The input data are the same of the DBAP formulation plus one:

- b_i: upper bound of the service time window on the ship i.

The problem is modeled on a multi-graph $G^k = (V^k, A^k), \forall k \in M$, where $V^k = N \cup \{o(k), d(k)\}$ and $A^k \subseteq V^k \times V^k$.

The following variables and constants are defined:

- $x_{ij}^k \in \{0,1\}$ $k \in M$, $(i,j) \in A^k$, $x_{ij}^k = 1$ if and only if ship j is scheduled after ship i at berth k;

- T_i^k $k \in M$, $i \in N$: berthing time of ship i at the berth k, i.e., the time when the ship moors (see Figure 2);
- $T_{o(k)}^k$ $k \in M$: starting operation time of berth k, i.e., the time when the first ship moors at the berth;
- $T_{d(k)}^k$ $k \in M$: ending operation time of berth k, i.e., the time when the last ship departs from the berth;
- $M_{ij}^k = \max\left\{b_i + t_i^k - a_j,\ 0\right\}, k \in M$, i and $j \in N$.

The MDVRPTW model is as follows:

$$\text{minimize} \sum_{i \in N} \sum_{k \in M} \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \right] \tag{9}$$

subject to

$$\sum_{k \in M} \sum_{j \in N \cup \{d(k)\}} x_{ij}^k = 1 \qquad \forall i \in N \tag{10}$$

$$\sum_{j \in N \cup \{d(k)\}} x_{o(k),j}^k = 1 \qquad \forall k \in M \tag{11}$$

$$\sum_{i \in N \cup \{o(k)\}} x_{i,d(k)}^k = 1 \qquad \forall k \in M \tag{12}$$

$$\sum_{j \in N \cup \{d(k)\}} x_{ij}^k - \sum_{j \in N \cup \{o(k)\}} x_{ji}^k = 0 \qquad \forall k \in M,\ \forall i \in N \tag{13}$$

$$T_i^k + t_i^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \qquad \forall k \in M,\ \forall (i,j) \in A^k \tag{14}$$

$$a_i \leq T_i^k \qquad \forall k \in M,\ \forall i \in N \tag{15}$$

$$T_i^k + t_i^k \sum_{j \in N \cup \{d(k)\}} x_{ij}^k \leq b_i \qquad \forall k \in M,\ \forall i \in N \tag{16}$$

$$s^k \leq T_{o(k)}^k \qquad \forall k \in M \tag{17}$$

$$T_{d(k)}^k \leq e^k \qquad \forall k \in M \tag{18}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall k \in M,\ \forall (i,j) \in A^k. \tag{19}$$

The objective function is always the minimization of the sum of the service times. When ship i is not assigned to berth k, the corresponding term in the objective function is zero because $\sum_{j \in N \cup d(k)} x_{ij}^k = 0$ and $T_i^k = a_i$, and the objective is minimized. Constraints (10) state that for every ship i there is exactly one active arc $(i,j) \in A^k$ $\forall k \in M$. Constraints (11) and (12) define the degree of the depots, while flow conservation for the remaining vertices is ensured by constraints (13). The consistency of the T_i^k variables with the sequence on the berth is achieved by constraints (14). The service time windows on the ships are set by constraints (15) and (16), and the availability time windows on the berths by constraints (17) and (18).

3 Tabu search heuristics

For small instances, both integer linear programming formulations of Section 2 can be solved by CPLEX. From a computational point of view, DBAP is better than MDVRPTW in that it can solve larger instances. However, neither model can be used for the optimal solution of instances of realistic size. The MDVRPTW formulation is in a sense more interesting than DBAP since it can easily accommodate time windows, and easily lends itself to the development of heuristics. We have developed such a heuristic, based on tabu search (Glover, 1986) for solving the discrete version of the BAP as a variant of the MDVRPTW. Since this heuristic focuses on the temporal dimension we call it T^2S (Time based Tabu Search). An extension of the tabu search heuristic to the spatial dimension is also presented for the continuous case. It is called $(TS)^2$ (Time and Space based Tabu Search).

3.1 T^2S - A tabu search heuristic for the discrete case

Our T^2S heuristic is inspired but different from the MDVRPTW algorithm of Cordeau et al. (2001). In the berth allocation problem, the function to be minimized is the sum for every ship of the service time in the port, as opposed to to the distance traveled in the MDVRPTW. Since the cost evaluation of the moves from a current solution is a recurrent step in the tabu search algorithm, some data structure were introduced to track the changes in the cost function in order to reduce the computational load. Other differences in the initialization phase and the insertion procedure will be explained in Section 3.1.2.

3.1.1 Notation and overview of the algorithm

Denote by S the set of BAP solutions that satisfy constraints (10) - (15) and (17). The heuristic explores the solution space by moving at each iteration from the current solution s to the best solution in its neighborhood $N(s)$. Each solution $s \in S$ is represented by a set of m berth sequences such that every ship belongs to exactly one sequence. This solution may, however, violate the time window constraints associated with the ships and the berths. The time window constraint on ship i on a berth k is violated if the completion time $T_i^k + t_i^k$ of the ship is larger than the time window's upper bound b_i. Berthing before a_i is not allowed; in other words, $T_i^k \geq a_i$. Similarly, the time window of berth k is violated when the completion time of a ship i assigned to berth k is larger than the berth time window's upper bound.

Let $c(s)$ denote the cost of solution, and let $w(s)$ denote the total violation of time window constraints, equal to the sum of the violations on the n ships and the m berths. Solutions are then evaluated by means of a penalized cost function $f(s) = c(s) + \alpha w(s)$, where α is a positive parameter. By dynamically adjusting the value of this parameter the relaxation mechanism facilitates the exploration of the search space and is particularly useful for tightly constrained instances.

The tabu search method is based on the definition of attributes used to characterize the solutions of S. They are also used to control tabu tenures and to implement a diversification

strategy. With each solution $s \in S$ is associated an attribute set $B(s) = \{(i,k)$: ship i is assigned to berth$\}$. The neighborhood $N(s)$ of a solution s is defined by applying a simple operator that removes an attribute (i,k) from $B(s)$ and replaces it with another attribute (i,k'), where $k \neq k'$. When ship i is removed from berth k, the sequence is simply reconnected by linking the predecessor and successor of the ship. Insertion in sequence k' is then performed between two consecutive ships so as to minimize the value of $f(s)$. When a ship i is removed from berth k, its reinsertion in that berth is forbidden for the next θ iterations by assigning a tabu status to the attribute (i,k).

An aspiration criterion allows the revocation of the tabu status of an attribute if that would allow the search process to reach a solution of smaller cost than that of the best solution identified having that attribute. To diversify the search, any solution $\overline{s} \in N(s)$ such that $f(\overline{s}) \geq f(s)$ is penalized by a factor proportional to the addition frequency of its attributes, and by a scaling factor. More precisely, let ρ_{ik} be the number of times attribute (i,k) has been added to the solution during the process and let ζ be the number of the current iteration. A penalty $p(\overline{s}) = \lambda c(\overline{s})\rho_{ik}/\zeta$ is added to $f(\overline{s})$. The scaling factor $c(\overline{s})$ introduces a correction to adjust the penalties with respect to the total solution cost. Finally, the parameter λ is used to control the intensity of the diversification. These penalties have the effect of driving the search process toward less explored regions of the search space. For notational convenience, assume that $p(\overline{s}) = 0$ if $f(\overline{s}) < f(s)$.

3.1.2 Description of the T^2S heuristic

The algorithm uses two different procedures to generate a starting solution. The first procedure, called R-G (random-greedy), places the ships at random in a queue. The first ship in the queue is assigned to the berth which allows the minimum completion time. For each subsequent ship an evaluation of all the possible assignments is performed and the insertion yielding the minimum cost increase is selected. The second procedure, called FCFS-G (first come, first served - greedy), orders the ships in the queue according to their arrival time. This queue is processed as in R-G. In both cases the initial solutions are constructed by relaxing the time window constraints. These initial solutions are usually infeasible, but this is not a problem for the tabu search heuristic.

The search starts from the solution produced by the R-G procedure and selects, at each iteration, the best non-tabu solution $\overline{s} \in N(s)$. After each iteration, the value of the parameter α is modified by a factor $1 + \delta$, where $\delta > 0$. If the current solution is feasible with respect to time window constraints, the value of α is divided by $1 + \delta$; otherwise, it is multiplied by $1 + \delta$. This process is repeated for η iterations and the best feasible solution s^* is updated throughout the search. The algorithm then restarts with the solution provided by the FCFS-G procedure. The complete algorithm can be summarized as follows:

1. Set $h := 1$.

2. Initialization phase: obtain a solution $s \in S$ using the R-G procedure, if $h = 1$; otherwise

use the FCFS-G procedure.

3. Set $\alpha := 1$. If s is feasible, set $s^* := s$ and $c(s^*) := c(s)$; otherwise, set $c(s^*) := \infty$.

4. For $j := 1, ..., \eta$, do

   (a) Select a solution $\overline{s} \in N(s)$ that minimizes $f(\overline{s}) + p(\overline{s})$ and is not tabu or satisfies its aspiration criterion.

   (b) If solution $\overline{s}$ is feasible and $c(\overline{s}) < c(s^*)$, set $s^* := \overline{s}$ and $c(s^*) := c(\overline{s})$.

   (c) Compute $w(\overline{s})$ and update α accordingly.

   (d) Set $s := \overline{s}$.

5. If $h = 1$, set $h := 2$, $s_1^* := s^*$ and go to Step 2.

6. Set $s^* := \min\{s_1^*, s^*\}$ and stop.

3.2 $(TS)^2$ - A tabu search heuristic for the continuous case

Since the T^2S heuristic works with a given set of berthing points the ship allocation to these points may not always satisfy the spatial constraints when the berthing points are too close to each other. Discrete models can be applied when the length of the quay is not a limiting factor for the terminal performance. This is not true for large transshipment terminals. To properly take ship lengths into account we have examined the ship length distribution obtained from the Gioia Tauro database (Figure 3). The average length is 183 meters, and the standard deviation is 71 meters. As can be easily observed, the maximum length is less than twice the average and the minimum length is almost equal to half the average.

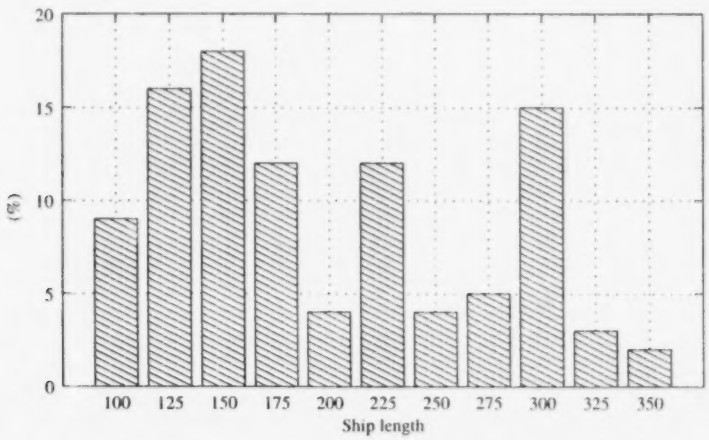

Figure 3: Distribution ship lengths

We therefore subdivide the quay into m berth segments where each segment k has a length L_k close to the ship length average. A segment k that is not the initial or the final berth segment

($k \neq 1$ and $k \neq m$) is then divided into two equal parts, left and right. Each segment has two neighbours, the right part of the segment $k-1$ and the left part of the segment $k+1$. Figure 4 shows an example. The initial and final segments are considered as special cases with only one neighbour and are not divided. These end segments can be used to model not only the beginning and the end of the quay, but also natural discontinuities on the quay, like sharp curves. In fact at Gioia Tauro there is such a discontinuity in the middle of the quay.

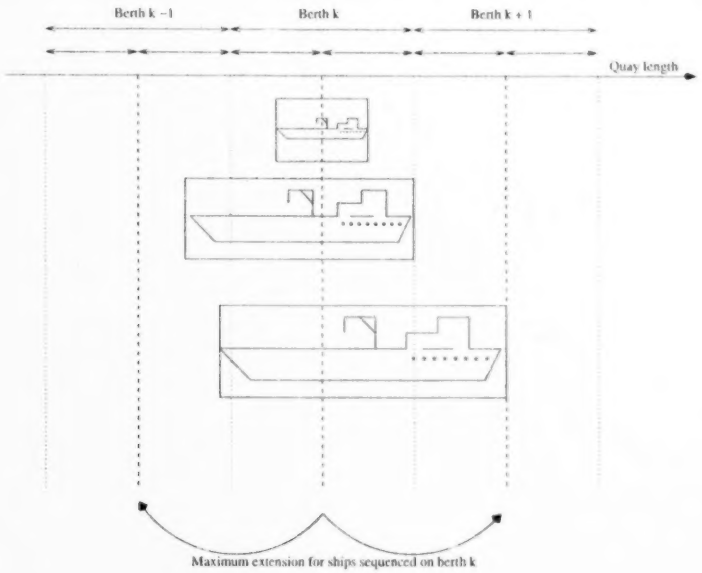

Figure 4: The quay as a set of segments with dynamically changing lengths

The length of a segment can be dynamically adjusted during the course of the algorithm. Thus a segment can accept a ship larger than its length, expanding itself toward one or the two of its neighbours. Conversely, when a segment accepts a ship smaller than its length, the unused space can be allocated to ships berthing in neighbouring segments. Of course the relation between the length of segments and ships matters. If ship lengths had been highly variable, our discretization approach would have caused a heavy fragmentation of the decision space.

3.2.1 Modifications to the T^2S heuristic

Representing the quay as a collection of variable length segments induces some changes in the tabu search algorithm. Every ship insertion or deletion in a berth sequence has an impact on the neighbouring berth segments. We now describe the new deletion and insertion procedures. A deletion procedure of ship i from berth k consists of the following steps:

1. Declare free the space occupied by ship i during its handling time; since a ship may also occupy part of neighbouring berths, the data structure relative to the these berths must

also be updated.

2. Iteratively call the deletion procedure for all ships that may follow ship i in berth sequence k.

3. Reinsert at minimum cost, using the insertion procedure explained below, all ships deleted in Step 2 in berth sequence k.

Let r be the number of ships already assigned to berth k. The insertion procedure for ship i in berth k evaluates all possible $r+1$ new sequences originated from the original one plus the progressive shifting of ship i to be inserted. Each new sequence then differs from the original one by its tail, i.e., the subsequence from ship i to the last ship in the original sequence. In the algorithm for the discrete case, which only took into account the temporal dimension, the evaluation of the $r+1$ new sequences was straightforward, involving only an adjustment of the berthing time of the ships in the tail sequence. This was accomplished by a single scanning of the tail sequence. Incorporating the spatial constraints and the interactions between neighbouring segments requires additional computations. For example, if ship i is inserted in position q, with $q \leq r$, then all ships in positions $q, .., r$ are deleted with the above mentioned procedure, meaning that their allocated space is declared free. Then a feasible space allocation is searched for the ship i and the deleted $r-q+1$ ships. For every ship the search for a feasible space allocation requires considering possible expansion to neighbouring segments and respecting, eventually, some earlier contractions that were required by the insertion procedure in neighbouring berths. The procedure looks for a feasible allocation that results in a good packing of ships in the two-dimensional space.

In $(TS)^2$ the initial solution is also built in a different way. The R-G procedure described in Section 3.1.2 was effective to avoid an initial local minimum, but with the spatial allocation the degree of fragmentation of this initial solution was important, requiring an important augmentation in the number of iterations to fix it. We therefore opted for the FCFS-G procedure. Note that this starting procedure is close to the behavior of the human planners. All tabu search mechanisms in $(TS)^2$ are the same as in the discrete case.

4 Computational results

We now present computational experiments for the discrete and continuous versions of the BAP. We first describe how realistic test instances were generated, and we then provide results obtained with the various algorithms.

4.1 Generation of test instances

In order to generate realistic instances, a statistical analysis of the traffic and berth allocation data at Gioia Tauro was performed (Legato et al., 2002). The ships served at the port are classified according to their function (feeders and mother vessels) and according to their port route.

A port route is the sequence of ports visited by a ship and is determined by the shipping company. There are 36 port routes for the feeders and 16 for the mother vessels.

A shipping company usually requests from the port management that areas of the yard and of the quay be dedicated to a subset of port routes. These are called *favorite* areas. This constraint is not always fully enforced in practice. In order to avoid traffic congestion a ship may be assigned to an *acceptable* area which is different from its favorite area. Port routes can be subdivided into six subsets. In Table 1, we provide two statistics on these subsets: the proportions of ships and of the container traffic. The quay is treated by the management as five areas, each dedicated to one or more port routes subsets and capable of accepting some others. In Table 2, we show the relationships between port routes subset and the quay areas in term of favorite and acceptable (indicated by "F" and "A"). To generate the instances we used a finer subdivision consisting of 13 berth segments. Berthing areas are not the only reason why ships are classified into different port route subsets. The statistical distribution of ship lengths and handling times also vary significantly between the various subsets. The generation of our test instances therefore takes into account various length and handling time distributions.

Subset	Description	Ships	Container traffic
		(% of total)	(% of total)
CF	Common feeder	36.7	19.1
DF	Dedicated feeder	27.5	31.8
O1	Mother vessel	9.2	24.6
O2	Mother vessel	6.3	7.4
O3	Mother vessel	14.3	11.4
O4	Mother vessel	6.0	5.7

Table 1: Port route subsets

Subset	Total quay length: 2850 meters				
	525 m.	700 m.	350 m.	350 m.	925 m.
CF	*F*	*A*	*A*	*A*	
DF				*A*	*F*
O1				*A*	*F*
O2	*A*	*F*	*A*		
O3		*A*	*F*		
O4		*A*	*A*	*F*	

Table 2: Subdivision of the Gioia Tauro quay into favorite and acceptable areas

The average number of ships arriving at Gioia Tauro is 60 per week (Figure 5). Since our exact formulation for the discrete case could not handle such a large value we worked with a smaller number of ships and a smaller quay length. For the discrete case five sizes were considered: 25 ships and 5, 7, 10 berths; 35 ships and 7, 10 berths. For each problem size, ten instances were randomly generated with different seeds.

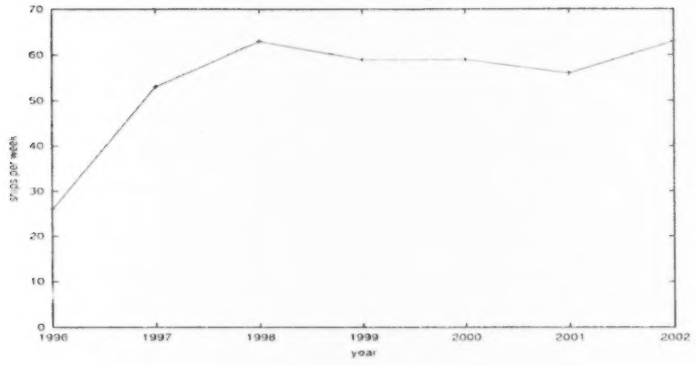

Figure 5: Ships' arrivals at Gioia Tauro port per week (yearly average)

The earliest available time of berth k, denoted by s^k, was generated as in Imai et al. (2001):

- s^k is the same for every berth;
- s^k is set for each instance equal to a given fraction f of the time duration between the first and last arriving ships.

The parameter f, equal to the expected proportion of ships present in the port when the berths become available, controls how much an instance is close to the "static" case. Thus, for $f = 1$ the DBAP reduces to the SBAP. The computational experiments carried out by Imai et al. (2001) show that the DBAP is easy to solve with a Lagrangean relaxation heuristic as long as f is large. Four different instance groups were defined by Imai with f equal to $1/2$, $5/8$, $3/4$ and $7/8$.

We carried out similar tests with the DBAP formulation implemented in CPLEX. Our instances are not the same as those of Imai et al. since we used the handling times observed at Gioia Tauro. Nevertheless, we reached conclusions similar to those of Imai et al. These results will be presented in the next subsection. The computational effort required by CPLEX to solve the problem is inversely related to f. Optimal solution can typically be obtained for $f \geq 1/2$. Unfortunately, when the berth allocation is solved with one week rolling horizon, realistic values for f are in the range $1/21, .., 1/7$. For our application f was fixed at $1/12$. But, in order to make the problem easier for CPLEX, four other classes were defined for instances involving 25 ships and 5 berths with the value of f used by Imai et al. We realize that these instances are not realistic; they were only generated to measure the performance of the heuristic. To summarize, we generated 90 instances subdivided into two subsets:

- I1: 40 instances with f equal to $1/2$, $5/8$, $3/4$ and $7/8$; problem size 25 ships and five berths,
- I2: 50 instances with $f = 1/12$ and with five different problem sizes.

These 90 instances were created for the DBAP formulation. Therefore the time windows associated with the ships have no upper bound.

4.2 Computational results for the discrete case

The DBAP formulation was implemented in CPLEX 7.1 with a time limit of two hours. Some sensitivity analyses to the CPLEX MIP solver parameters were carried out, but the best results were obtained with the default values of these parameters. The tabu search algorithm was implemented in ANSI C. Computational experiments were performed on a SUN workstation (900 MHz). Table 3 provides average results obtained on each instance size. The instances of subset I1 are easily solved by CPLEX and T^2S, which are almost always able to reach the optimum. However, this subset was only used as a first benchmark. The more challenging instances are those of subset I2. Here CPLEX could not reach an optimal solution within the time limit. We therefore used a truncated branch-and-bound which stops with the best feasible solution identified after two hours. In contrast T^2S was capable of identifying, within a few seconds, a much better solution than the truncated CPLEX algorithm. The number of constraints, variables and non-zero coefficients in the reduced MIP solved by CPLEX are reported in Table 4 for each instance size.

The iteration count limit η in the tabu search heuristic was set equal to 10^4, but in the vast majority of the instances the best solution was identified before 10^3 iterations. The tabu search parameters used for these computations are as follows:

- θ : tabu duration equal to $\lfloor 7.5 \log n \rfloor$;
- λ : diversification intensity parameter equal to 0.015;
- δ : penalty adjustment parameter equal to 0.5.

These values are the same used in Cordeau et al. (1997) and in Cordeau et al. (2001). We provide in Tables 5 and 6 results for the I2 instances with additional statistics (number of nodes explored in the branch-and-bound tree rounded to the nearest multiple of 100). The advantage of this heuristic with respect with the truncated CPLEX algorithm is clear from these results: T^2S is not only much faster, but it also identifies much better solutions than the truncated CPLEX algorithm.

		Truncated CPLEX			T^2S		T^2S vs CPLEX
Problem Size (ships x berths)	$f^{(a)}$	Objective Value	Gap$^{(b)}$ (%)	CPU (s)	Objective Value	CPU (s)	Improvement (%)
25 x 5	1/2	1722	0.3	2809	1722	6	-
25 x 5	5/8	2187	0.0	188	2187	6	-
25 x 5	3/4	2551	0.0	39	2551	6	-
25 x 5	7/8	2877	0.0	19	2877	6	-
25 x 5	1/12	1058	49.0	7200	879	4	17
25 x 7	1/12	990	87.1	7200	745	4	25
25 x 10	1/12	1009	109.9	7200	766	4	24
35 x 7	1/12	2040	68.1	7200	1236	8	39
35 x10	1/12	2036	83.7	7200	1188	6	42

Table 3: Summary of the average results (10 instances for every problem size)

a - fraction of the time duration between the first and last arriving ship.

b - The gap is calculated with respect to the value of the linear relaxation as (upper bound - lower bound)/upper bound.

Problem size (ships x berths)	Constraints	Variables	Non-zero coefficients
25x5	2870	5885	1,493,055
25x7	3972	8233	2,068,323
25x10	5580	11485	2,843,955
35x7	7634	15985	7,979,899
35x10	10979	22474	11,329,854

Table 4: Number of constraints, variables and non-zero coefficients in the reduced MIP solved by CPLEX

4.3 Computational results for the continuous case

In the continuous case an exact formulation developed by Kim and Moon (2003) is capable of solving only very small size instances. Therefore, the results of the $(TS)^2$ heuristic are evaluated against two benchmarks:

- The final solution of the T^2S heuristic applied on the same input data, but of course, discarding the spatial constraints. The discrete case can be regarded as a relaxation of the continuous case since the solution of the T^2S heuristic may be infeasible with respect to the spatial constraints.
- The starting solution, provided by the FCFS-G procedure. This comparison estimates the possible savings that the heuristic can provide to the port with respect to a manual plan.

Thirty instances were randomly generated. Computational results are provided in Table 7. The $(TS)^2$ heuristic finds within 120 seconds solutions which are, on the average, 8% better than those obtained by the FCFS-G procedure. This improvement is highly significant in practice.

		Truncated CPLEX			T^2S	T^2S vs CPLEX
Problem size (ships x berths)	Instance	Objective value	Gap [a] (%)	Number of nodes [b]	Objective value	Improvement (%)
25 x 5	1	846	29.5	23700	759	11
	2	1244	53.7	2300	965	29
	3	1033	12.9	15700	974	6
	4	946	75.3	9300	702	35
	5	1082	37.0	15600	965	12
	6	1255	31.4	16200	1129	11
	7	1039	57.6	7500	835	24
	8	843	85.2	8800	629	34
	9	1066	74.2	9700	755	41
	10	1227	32.7	8400	1077	14
25 x 7	1	869	90.1	1900	667	30
	2	918	94.1	4000	671	37
	3	971	52.2	7200	823	18
	4	983	98.3	1400	655	50
	5	948	78.5	2700	728	30
	6	1175	107.4	1000	794	48
	7	1094	115.8	1000	740	48
	8	939	64.4	3600	782	20
	9	987	75.6	3700	759	30
	10	1018	94.6	2700	830	23
25 x 10	1	900	86.8	1900	717	26
	2	957	103.6	1800	736	30
	3	1135	119.7	300	764	49
	4	1082	92.6	900	819	32
	5	1010	79.4	1800	855	18

Table 5: Computational results for truncated CPLEX and T^2S, part I

a - The gap is calculated with respect to the value of the linear relaxation as (upper bound - lower bound)/upper bound.

b - Number of nodes in the branch-and-bound tree rounded to the nearest multiple of 100.

5 Conclusions and future developments

We have provided two formulations for the discrete version of the BAP and we have developed two heuristics, one for the discrete case and one for the continuous case. The solution values provided by the first was compared on small instances to the optimal values generated by the exact DBAP formulation. On these instances our heuristic for the discrete case (T^2S) always yields optimal solutions. On larger instances it always outperforms the truncated CPLEX algorithm applied to the DBAP. Since the discrete model does not always satisfy the spatial constraints, we have developed another heuristic $(TS)^2$ for the continuous case. Comparisons were made with the discrete case and with a simple constructive procedure. Both heuristics are capable of solving more realistic instances than those previously considered by other authors. Our two heuristics can handle the various features of real-life problems, including time windows, favorite and acceptable berthing areas, etc. The objective function can easily accom-

		Truncated CPLEX			T^2S	T^2S vs CPLEX
Problem size (ships x berths)	Instance	Objective value	Gap[a] (%)	Number of nodes[b]	Objective value	Improvement (%)
25 x 10	6	919	138.0	900	694	32
	7	867	124.0	900	673	29
	8	1046	108.7	1600	860	22
	9	1141	138.3	500	726	57
	10	1032	107.5	1200	812	27
35 x 7	1	1606	56.0	800	1019	58
	2	2050	73.3	520	1196	71
	3	2245	67.6	500	1230	83
	4	1778	63.0	700	1150	55
	5	1962	77.5	600	1179	66
	6	2496	50.6	500	1703	47
	7	1494	69.1	600	1181	27
	8	2582	68.8	500	1330	94
	9	2244	85.3	300	1245	80
	10	1941	69.4	700	1130	72
35 x 10	1	1578	90.1	300	1128	40
	2	1854	70.5	400	1197	55
	3	1667	110.9	70	953	75
	4	2284	93.2	70	1239	110
	5	2739	75.3	60	1372	102
	6	1956	79.6	300	1221	62
	7	2124	86.7	0	1052	102
	8	2133	100.9	110	1219	75
	9	1668	44.1	900	1315	27
	10	2001	86.0	280	1198	67

Table 6: Computational results for truncated CPLEX and T^2S, part II

a - The gap is calculated with respect to the value of the linear relaxation as (upper bound - lower bound)/upper bound.

b - Number of nodes in the branch-and-bound tree rounded to the nearest multiple of 100.

modate average container loading and unloading times which are dependent on the berthing position, leading to a bicriterion optimization problem. The integration of the berthing and quay cranes assignment problems will be the object of a further study. The port of Gioia Tauro plans to incorporate our heuristics in their decision support system.

Acknowledgments

This work was supported by the Canada Research Chair in Distribution Management, by the HEC Centre for International Business Studies, by the CN Chair in Transport Economics and Intermodality, and by a strategic research grant from HEC Montréal. The authors thank the Gioia Tauro port managers Carmine Crudo and Giuseppe Perri for their kind availability, and Flavia Monaco for her valuable comments. Special thanks are due to Manlio Gaudioso who initiated the cooperation framework between Gioia Tauro and Università della Calabria; this

Instance	T^2S Objective value	$(TS)^2$ Objective value	Comparison[a] (%)	FCFS - G Objective value	Improvement[b] (%)
1	1415	1706	21	1899	10
2	1263	1355	7	1417	4
3	1139	1286	13	1349	5
4	1303	1440	11	1548	7
5	1208	1352	12	1449	7
6	1262	1565	24	1747	10
7	1279	1389	9	1482	6
8	1299	1519	17	1616	6
9	1444	1713	19	1873	9
10	1212	1411	16	1611	12
11	1378	1696	23	1851	8
12	1325	1629	23	1814	10
13	1360	1519	12	1575	4
14	1233	1369	11	1435	5
15	1295	1455	12	1609	10
16	1375	1715	25	1854	7
17	1283	1322	3	1388	5
18	1346	1594	18	1923	17
19	1370	1673	22	1829	9
20	1328	1450	9	1615	10
21	1346	1565	16	1640	5
22	1332	1618	21	1747	7
23	1266	1539	22	1770	13
24	1261	1425	13	1625	12
25	1379	1590	15	1845	14
26	1330	1567	18	1707	8
27	1261	1458	16	1588	8
28	1365	1550	14	1669	7
29	1282	1415	10	1512	6
30	1351	1621	20	1797	10
Average	1310	1517	16	1659	8

Table 7: Computational results for $(TS)^2$
a - Comparison between the solutions provided by $(TS)^2$ and T^2S, calculated as $((TS)^2 - T^2S)/T^2S$.
b - Improvement of the T^2S heuristic with respect to the FCFS-G procedure.

research would not have been possible without his leading role.

References

[1] Christiansen M, Fagerholt K and Ronen D (2003). Ship routing and scheduling: Status and perspectives. *Transportation Science*. Forthcoming.

[2] Cordeau JF, Gendreau M and Laporte G (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30**: 105-119.

[3] Cordeau JF, Laporte G and Mercier A (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* **52**: 928-936.

[4] Daganzo C F (1990). The productivity of multipurpose seaport terminals. *Transportation Science* **24**: 205-216.

[5] De Castilho B, Daganzo C F (1993). Handling strategies for import containers at marine terminals. *Transportation Research* **27B**: 151-166.

[6] Garey M, Johnson D (1979). *Computers and Intractability*. Freeman, San Francisco.

[7] Glover F (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* **13**: 533-549.

[8] Kim K H, Moon K C (2003). Berth scheduling by simulated annealing. *Transportation Research* **37B**: 501-594.

[9] Imai A, Nagaiwa K, Chan W T (1997). Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation* **31**: 75-94.

[10] Imai A, Nishimura E and Papadimitriou S (2001). The Dynamic Berth Allocation Problem for a container Port. *Transportation Research* **35B**: 401-417.

[11] Imai A, Nishimura E and Papadimitriou S (2003). Berth allocation with service priority. *Transportation Research* **37B**: 437-457.

[12] Legato P, Monaco F and Tigani N (2001). Berth planning at Gioia Tauro's maritime terminal by logistic distribution models. AIRO Annual Conference, Cagliari, Italy.

[13] Legato P, Monaco F and Sammarra M (2002). Il problema del berth allocation. Technical report n. 2, Laboratorio di Logistica, Università della Calabria, Italy.

[14] Lim A (1998). The berth planning problem. *Operations Research Letters* **22**: 105-110.

[15] Nishimura E, Imai A and Papadimitriou S (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research* **131**: 282-292.

[16] Papadimitriou C H, Steiglitz K (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ.

[17] Park Y M, Kim K H (2003). A scheduling method for berth and quay cranes. *OR Spektrum* **25**: 1-23.

[18] Pinedo M (1995). *Scheduling: Theory, Algorithms and Systems*. Prentice-Hall, Englewood Cliffs, NJ.

[19] Taleb-Ibrahimi M, De Castilho B, Daganzo C F (1993). Storage space versus handling work in container terminals. *Transportation Research* **27B**: 13-32.

[20] Vis I F A, De Koster R (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research* **147**: 1-16.

[21] World Bank (2003). http://www.worldbank.org/transport.